8-10-2004

A Distributed Evolutionary Algorithmic Approach to the Least-Cost Connected Constrained Sub-Graph and Power Control Problem

Jason C. Tillett
Rochester Institute of Technology

Raghuveer Rao
Rochester Institute of Technology

Ferat Sahin
Rochester Institute of Technology

T. M. Rao
SUNY Brockport

Recommended Citation

Jason C. Tillett, Raghuveer Rao, Ferat Sahin, T. M. Rao, "A distributed evolutionary algorithmic approach to the least-cost connected constrained sub-graph and power control problem", Proc. SPIE 5440, Digital Wireless Communications VI, (10 August 2004); doi: 10.1117/12.541663; https://doi.org/10.1117/12.541663

A distributed evolutionary algorithmic approach to the least-cost connected constrained sub-graph and power control problem

Jason Tillett[a] , Raghuveer Rao[a], Ferat Sahin[a], and T. M. Rao[b]
[a]Rochester Institute of Technology, Rochester NY 14623
[b]SUNY at Brockport Brockport, NY

ABSTRACT

When wireless sensors are capable of variable transmit power and are battery powered, it is important to select the appropriate transmit power level for the node. Lowering the transmit power of the sensor nodes imposes a natural clustering on the network and has been shown to improve throughput of the network. However, a common transmit power level is not appropriate for inhomogeneous networks. A possible fitness-based approach, motivated by an evolutionary optimization technique, Particle Swarm Optimization (PSO) is proposed and extended in a novel way to determine the appropriate transmit power of each sensor node. A distributed version of PSO is developed and explored using experimental fitness to achieve an approximation of least-cost connectivity.

1. INTRODUCTION

A common theme in energy-aware sensor network operation is that the network lifetime is extended by reducing the power output of the nodes. However, reducing the transmit power of a wireless device may not translate into real energy savings. This is because the total power required to transmit a packet of data must include the additional power required to retransmit it when the packet is not received correctly at the destination because of collisions. Therefore, finding an "optimum" transmit power for each node is a constrained optimization problem whose objective function must be evaluated in the context of the sensor network application. It has been demonstrated[1] that the optimum transmit power of the nodes varies with load on the network, assuming a common transmit power. Here, we allow each node to have a different transmit power, but we address the same problem of finding the optimal transmit power of the sensor nodes. By optimal, we mean that the topology generated by the algorithm will perform well in terms of throughput for a wide range of network loads using a contention-based communication protocol.[2] Our optimization specifically targets contention introduced by hidden nodes and asymmetric links. This is not the case for most topology/power control algorithms. However, there are exceptions.[3] In section 2, we formulate a statement of the constrained optimization problem. The PSO algorithm is described briefly in section 3. A distributed extension of the PSO algorithm is proposed and developed generally in section 4. The subsections of section 4 are used to cast the distributed algorithm in a form suitable for attacking the problem statement. Section 5 details the simulations and results. A summary and future work section concludes the paper.

2. PROBLEM STATEMENT

Part of the problem is to find a connected[i] group of N graph vertices that minimizes the cost of connectivity. [4] We assume all edges/links have symmetric costs. Without constraints, this is just the minimum cost spanning tree problem. [5] We add the constraint that if a retained edge emanating from a node has weight *w*, then all other possible edges emanating from the node that have weight less than or equal to *w* should also be included in the graph. The purpose of the constraint is to encourage symmetric links. Symmetric links are needed for proper functioning of many protocols across different layers of the protocol stack. The MAC layer relies on symmetric links for acknowledgements and many routing protocols assume symmetric links. [6-8] This constrained optimization problem can be expressed as follows. Let $\vec{r}$ represent the communication radii of the nodes. Define $C(\vec{r})$ to be a function that returns 1 if the nodes are connected. Let Ω_i be the set of node indexes that are able to correctly receive packets from node i. Given that d_{ij} is the distance between nodes i and j, the problem statement is,

[i] There exists a path from any node to any other node in the graph. Since our graphs are not directed, connected is identical to strongly-connected.

Digital Wireless Communications VI, edited by Raghuveer M. Rao, Sohail A. Dianat,
Michael D. Zoltowski, Proceedings of SPIE Vol. 5440 (SPIE, Bellingham, WA, 2004)
0277-786X/04/$15 · doi: 10.1117/12.541663

$$
\begin{aligned}
&\min(\sum_{i=1}^{N} r_i) \\
&\textit{subject to,} \\
&C(\vec{r}) = 1 \qquad\qquad . \\
&\vec{r}_{\min} < \vec{r} < \vec{r}_{\max} \\
&\left\{ d_{ij} < r_j \mid j \in \Omega_i\left(\vec{r}\right) \right\}
\end{aligned} \tag{1}
$$

In Eqn. 1, for clarity, we have explicitly included the dependence of Ω_i on the current communication radii of the nodes.

This problem statement can easily be justified as perfectly applicable to a sensor network. In a sensor network, each node/vertex is assumed to be a wireless sensor with limited radio communication range. In general, a node will not be able to reach all of the other nodes in the sensor network. It has been shown that for a uniformly distributed wireless network, throughput and energy consumption can be optimized by reducing the common power level of the nodes. [9, 10] Therefore, we want to reduce the communication range of all of the nodes, just to the point where they are all still connected. In this configuration, to minimize collisions, links should be maintained with all nodes that are within a node's communication range. This is equivalent to minimizing the number of asymmetric links. Although a global parameter optimization is required, nodes may not have global knowledge, and therefore must participate in solving the problem using only local interactions.

3. THE APPROACH: PARTICLE SWARM OPTIMIZATION (PSO)

The PSO [11] approach utilizes a cooperative swarm of particles, where each particle represents a candidate solution, to explore the space of possible solutions to the optimization problem of interest. Each particle is randomly or heuristically initialized and then allowed to 'fly'. At each step of the optimization, each particle is allowed to evaluate its own fitness and the fitness of its neighboring particles. The fitness or objective function is a function of the solution. Each particle can keep track of its own solution, which resulted in the best fitness, as well as see the candidate solution for the best performing particle in its neighborhood. At each optimization step, each particle adjusts its candidate solution (flies) according to,

$$
\begin{aligned}
v(t+1) &= v(t) + \phi_1(x - x_p) + \phi_2(x - x_n) \\
x(t+1) &= x(t) + v(t+1)
\end{aligned} \tag{2}
$$

Subscripts for particle index and dimensionality have been left off of Eqn. 2, which may be interpreted as the 'kinematic' equation of motion for one of the particles (test solution) of the swarm where the particle is one-dimensional. The variables in Eqn. 2 are summarized in Table 1.

Table 1- List of variables used in the equations

v	The particle velocity.
x	The particle position (test solution).
t	Time
ϕ_1	A uniform random variable usually distributed over [0,2].
ϕ_2	A uniform random variable usually distributed over [0,2].
x_p	The particle's position (previous) that resulted in the best fitness so far.
x_n	The neighborhood position that resulted in the best fitness so far.

Eqn. 2 can be interpreted as follows. Particles combine information from their previous best position and their neighborhood best position to maximize the probability that they are moving toward a region of space that will result in a better fitness.

The application of PSO to solve the minimum cost constrained sub-graph problem we have described is not trivial since each sensor node must select its power level / communication range autonomously. There is no communication with a base station that can perform the optimization and pass along optimal communication ranges to each of the nodes. Thus, even if we could construct a PSO particle representing the solution and construct an appropriate fitness, the sensor network could not compute the result. We need a distributed version of the PSO algorithm.

4. DISTRIBUTED PSO (DPSO)

In traditional PSO, the fitness function is shared among the particles in the swarm. Particles in traditional PSO represent the solution to a single optimization problem. In contrast, in the distributed form developed here, particles have no knowledge, or limited knowledge, of the global objective function. Particles do not represent a global solution to a single optimization problem. Rather, particles have individual objectives and their objective function is a function of their individual parameters. This can be written as,

$$f_i\left(p_{i1}, p_{i2}, \ldots p_{iM},\right). \tag{3}$$

where each particle, i, has M parameters. The operational parameters, p_{ij}, of Eqn. 3 can be: communication range, sensing range, carrier sense range, number of neighbors, battery reserve level. This list provided is exemplary and is neither complete nor the list used in this paper. The system designer may and probably will have a global objective or optimization targeted, like in Eqn. 1, but the particles cannot evaluate the global objective function because they do not have access to all of the nodes' communication radii, again addressing Eqn. 1 for illustration. It is up to the designer to craft a suitable local objective function that will cause the system to approximate the desired global objective.

Another difficulty arises due to the use of local objective functions and their dependence on local parameters. How does one calculate the neighborhood best (labeled x_n in Eqn. 2)? In traditional/centralized PSO, the neighborhood best is simply the solution represented by the most fit neighbor. In DPSO, the solution represented by the most fit neighbor evaluated using the particle's local objective function, may not, and probably won't, result in a better fitness for the particle. Therefore particles must be able to interpret solutions/parameters received from their neighbors. Particles must be able to convert parameter values and fitnesses exchanged with neighbors into a possible "better fit" set of values for their own operational parameters, their neighborhood best parameters. This mapping is expressed as $\tilde{Q}_i \Rightarrow \vec{p}_{i,nbest}$ where

$$\tilde{Q}_i = \begin{Bmatrix} (f_1, p_{11}, p_{12} \cdots p_{1M}), \\ (f_2, p_{21}, p_{22} \cdots p_{2M}), \\ \cdots \\ (f_{N_i}, p_{N_i 1}, p_{N_i 2} \cdots p_{N_i M}) \end{Bmatrix} \tag{4}$$

and $\vec{p}_{i,nbest}$ is the neighborhood best values for node i. $\tilde{Q}_i$ is a set, for node i, of values of fitnesses, f, and parameters, p, that it receives from its N_i neighbors. In Eqn. 4, each neighbor has a single fitness value, but may have up to M parameter values to report to particle i. Once each particle is able to evaluate its fitness function and is able to construct its $\tilde{Q}$ set with information from its neighbors, the computation can proceed as in traditional PSO. Eqn. 2 becomes

$$\begin{aligned} \vec{v}_i(t+1) &= \vec{v}_i(t) + \phi_1(\vec{p}_i - \vec{p}_{i,best}) + \phi_2(\vec{p}_i - \vec{p}_{i,nbest}) \\ \vec{p}_i(t+1) &= \vec{p}_i(t) + \vec{v}_i(t+1) \end{aligned}. \tag{5}$$

We have used slightly modified notation in Eqn. 5 in order to remove some labeling ambiguity. The subscript *best* denotes the previous best value for the particle, which determines the cognitive component of the particles' motions. The subscript *nbest* demotes the neighborhood best and determines the social component of the particles' motions. We changed x's in Eqn. 2 to p's to be consistent with the notation in Eqns. 3 and 4. We emphasize here that the calculation of the neighborhood best must be discussed in the context of a specific problem.

4.1 DPSO for the least-cost connected constrained sub-graph problem, the local fitness function

For applying DPSO to wireless sensors, we take the sensors/nodes to be the particles in the swarm, and a node's neighborhood consists of the set of nodes with which it has communication links. For approximating the least-cost, connected, constrained sub-graph problem, we must identify the fitness function to be used by the sensor nodes. This involves identifying the form as well as the parameters of each sensor node's fitness function. Also to be determined is an appropriate choice for $\tilde{Q}_i \Rightarrow \vec{p}_{i,nbest}$. First we identify some assumptions. We assume each node knows its position so that communicating nodes can calculate distances. We also assume that the sensor node can adjust its power so it can vary its reach or communication range. Define S_i to be the number of distinct nodes reachable by node i, including itself, and assume that a protocol is in place to build S_i for each node in each step of the DPSO algorithm. We leave to future work implementing a distributed representation of this 'reachability', S_i but note that a probabilistic measure, representing the probability that the network is strongly connected could be adopted. [12] We assume that each node communicates with all of its neighbors with the same power and does not vary its power on a neighbor-by-neighbor basis. [13] We assume that each node's carrier sense range is the same as its communication range. It is this assumption that will allow us to equate minimization of hidden nodes to minimization of neighbors for uniform node distributions. If the carrier sense range is not the same as its communication range, then a formal definition of a hidden node is,

$$\begin{gathered} d_{ij} \le r_i \,,\, d_{ij} \le r_j \,,\, i \ne j \\ d_{jk} \le r_j \,,\, d_{jk} \le r_k \,,\, d_{ik} > \frac{R_{sense}}{R_{receive}} r_k \,,\, i \ne k \end{gathered} \tag{6}$$

In Eqn. 6, node k is hidden from node i.

Power consumption has two main components. The transmit power of the node is the first component. The second component is the power consumed in re-transmitting frames lost due to collision at the MAC layer. Building minimization of power expended in normal transmissions into the fitness function is straightforward. We simply make the sensor node fitness proportional to the transmit radius, r_i. (We are minimizing fitness.) Minimizing the power expended in re-transmissions can be achieved by minimizing the impact of the hidden node problem. Both, reducing the number of hidden nodes, and maximizing the number of symmetric links can reduce the impact of hidden nodes. In a uniformly distributed collection of sensor nodes, the number of hidden nodes (2 hop neighbors) can be minimized by minimizing either the transmit radius or number of neighbors for a given node. Therefore in a general expression for the fitness of a node, we will make the fitness proportional to the transmit radius and proportional to the number of neighbors, $N_{1,i}$. If we denote $N_{2,i}$ as the number of nodes that a given node can "hear", but cannot reach in a single hop, then the number of asymmetric links can be minimized by minimizing $N_{2,i}$. Therefore, a general expression for the sensor node's fitness will be proportional to $N_{2,i}$. A node is most fit when it may send a packet successfully to any other node in the network. Therefore the fitness of a node will be inversely proportional to S_i. Finally, a general expression of each node's fitness is (node i)

$$f_i = \frac{\left(N_{1,i} + c_1\right)^{\alpha} \left(N_{2,i} + c_2\right)^{\beta} r_i^{\gamma}}{S_i^{\delta}} . \tag{7}$$

In Eqn. 7, the parameters, $\alpha, \beta, \gamma, \delta, c_1, c_2$ are constants to be determined. The fitness of Eqn. 7 is not applicable for all nodes. For solitary or disconnected nodes, an expression for the fitness which encourages disconnected nodes to broadcast with greater and greater power in an attempt to gain connectivity to a supposed existing sensor network is given by,

$$f_i(disconnected\ node\ i) = \frac{K}{r_i}\ , \tag{8}$$

where K is a large constant and r_i is the sensor node's communication range. Eqn. 8 expresses that solitary nodes are more fit when they expand their communication range. Through the discussion above, we have identified the parameters, $\vec{p}$, in the fitness function. They are N_1, N_2, r and S. Note that more parameters may be necessary and will be introduced as needed.

$$p_i = \left(r_i, N_{1,i}, N_{2,i}, S_i\right) \tag{9}$$

In Eqn. 9, $p_{i,1} = r_i$, $p_{i,2} = N_{1,i}$, $p_{i,3} = N_{2,i}$ and $p_{i,4} = S_i$. We argue that the set $\left\{p_{ij} \mid i = 1,2...N\ ,\ j = 1,2,3,4\right\}$ that minimizes $\vec{f}$ component by component, will result in a locally optimal, probably non-pareto solution to the least-cost, strongly-connected, constrained sub-graph problem. Specifically, if $\left\{S_i = N\right\} \cup \left\{N_{i2} = 0\right\}\ \forall i$, then the graph will be strongly-connected and all links will be symmetric.

The only parameter in the node's fitness function over which it has independent control is its communication radius, r_i. Therefore, Eqn. 5 can be re-written as,

$$\begin{aligned} \vec{v}_i(t+1) &= \vec{v}_i(t) + \phi_1(r_i - r_{i,best}) + \phi_2(r_i - r_{i,nbest}) \\ r_i(t+1) &= r_i(t) + \vec{v}_i(t+1) \end{aligned} \quad . \tag{10}$$

When the nodes fix their communication range, then the topology is fixed and N_{1i}, N_{2i} and S_i are determined. The only quantity left to discuss in Eqn. 10 is the neighborhood best. We hope to explain our interpretation of the neighborhood best in the DPSO approach and discuss explicitly how it is calculated. Again, this is one of the major challenges to applying this algorithm.

4.2 DPSO for the least-cost connected constrained sub-graph problem, the neighborhood best mapping

Since a node cannot simply query the most fit node for the value of $r_{i,nbest}$, it must infer the value. To maintain the spirit of cooperative swarming, the node must somehow use neighborhood information to infer $r_{i,nbest}$. How does the neighborhood indicate to a node what its $r_{i,nbest}$ should be? To start to understand the answer to this question, consider a node with no neighbors. It has a neighborhood, a null neighborhood. That neighborhood is also conveying information to the node. That information is "you are solitary and should increase your communication range". We have already implicitly built in this neighborhood influence right into the fitness function. We stated that when a node becomes disconnected, its fitness must change to cause it to expand its communication radius. Hence when $\tilde{Q} = 0$ (null set), the node should adjust its neighborhood best $r_{i,nbest}$ to a larger value. We are free to experiment with different ways to adjust the neighborhood best $r_{i,nbest}$ value in this situation. We could select the maximum communication range of the node, for example. This illustrates our interpretation of how the neighborhood can allow a node to compute a value for $r_{i,nbest}$. Now we discuss in more detail the neighborhood best for non-null neighborhoods.

4.2.1 Neighborhood best: 2-node experiments

As a first guess at an appropriate value for the neighborhood best $r_{i,nbest}$, nodes are encouraged to maintain connectivity with the worst fit node whose transmissions are detectable (packets can be correctly decoded). If a node is the worst fit in its detectable neighborhood, then it set its neighborhood best $r_{i,nbest}$ to be the distance to its nearest neighbor plus a small overlap. The reasoning behind this choice is that nodes will be encouraged to reach out to disconnected nodes and thereby promote connectivity. This definition of the calculation of $r_{i,nbest}$ can be represented as a simple state machine. The node internalizes 2 different states for $r'_{i,nbest}$. The prime indicates that the variable is different from $r_{i,nbest}$. One state, say $r'_{i,nbest}$ =0 means the nodes neighborhood best is set to the distance between it and its nearest neighbor and $r'_{i,nbest}$ =1 means that the node sets its neighborhood best $r_{i,nbest}$ to the distance between itself and its least fit detectable neighbor. If we define a variable U that represents the state on the environment, where $U = 0$ means that the node is the worst fit in its neighborhood and $U = 1$ means it is not, then the simple state machine is shown in Figure 1.

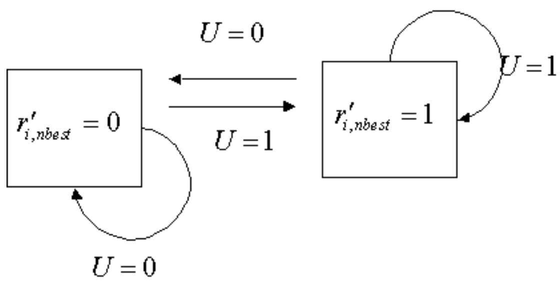

Figure 1 -- 2 node state machine to determine neighborhood best

The state machine of Figure 1 can be used to calculate $r_{i,nbest}$ at each step of the algorithm.

4.2.2 Neighborhood best: 3-node experiments

Unfortunately, when a third node is added, the algorithm cannot converge to an optimum solution for all initial conditions hence the state machine of Figure 1 is not sufficient for more than 2 nodes. For example, if all 3 nodes are initialized with a minimum communication range, then all 3 will begin expanding their communication ranges. The 2 nearest nodes will discover each other and stop exploring. As the communication range of the third node expands further, the 2 connected nodes will overhear the 3rd node and set their neighborhood best $r_{i,nbest}$ to the distance to the newly discernible, disconnected, and therefore unfit node. Both of the connected nodes will begin to expand their communication ranges in an attempt to gain connectivity with the orphaned node. After their communication ranges are sufficient to connect to the orphaned node, they are unable to determine which node should stay connected to the new node and which should contract its communication radius and allow the other to provide connectivity. Allowing each node to maintain a gateway status parameter can solve this problem. This will be another parameter that will be used by nodes for performing their $\tilde{Q}_i \Rightarrow \vec{p}_{i,nbest}$ mapping.

For 3 nodes we expand the state space of the node by introducing more states into $r'_{i,nbest}$ and defining a new state variable. We allow nodes to take on gateway roles. A new state variable, gw_i represents the gateway status of a node. Each node must somehow infer its ‘next state’ where its next state is a combination of its gateway status, gw_i and its neighborhood best communication range $r'_{i,nbest}$. gw_i can be 0 or 1, and $r'_{i,nbest}$ can be 0,1 or 2. These state values and their meanings are summarized in
Table 2.

Table 2 – node state variables relating to neighborhood best FSM (finite state machine), 3-node simulations

$r'_{i,nbest}$ =0	set $r_{i,nbest} = r_{\max}$
$r'_{i,nbest}$ =1	set $r_{i,nbest}$ = distance to nearest neighbor
$r'_{i,nbest}$ =2	set $r_{i,nbest}$ = distance to farthest neighbor
gw_i =0	node is not a gateway
gw_i =1	node is a gateway

Of the 6 possible combinations of the allowed internal states summarized in Table 2, we configure the transition function such that only 4 of the states are accessible. The transition function that determines the next state of the node, and hence its $r_{i,nbest}$, is constructed to be dependent on the factors that should affect a node's $r_{i,nbest}$. Specifically, the presence of asymmetric links or the lack of neighbors should encourage a node to explore. If a node is directly linked to a gateway, then resolving asymmetric links should be relegated to the gateway. We therefore identify 3 environmental states that factor into the transition function of the nodes FSM. They are the number of neighbors, the number of asymmetric links, and whether the node is linked to a gateway. Each of these states we allow to take on values 0 and 1. The states and their meanings are summarized in Table 3. If we represent the triple of these states as [$N'_1, N'_2, l2gw$], we can write the set of possible states as

Table 3 – environmental states in the FSM

N'_1 =0	node has no neighbors
N'_1 =1	node has one or more neighbors
N'_2 =0	node has no asymmetric links
N'_2 =1	node has one or more asymmetric links
$l2gw$ =0	node is not linked to a gateway
$l2gw$ =1	node is linked to a gateway

$$U_1 \therefore \left\{ \begin{matrix} [0,0,0] \\ [1,0,0] \\ [0,0,1] \\ [1,0,1] \end{matrix} \right\}, U_2 \therefore [1,1,1], U_3 \therefore [1,1,0], U_4 \therefore [0,1,1], U_5 \therefore [0,1,0].$$

With these definitions, we can provide visualization for the full FSM for simulations with 3 nodes in Figure 2.

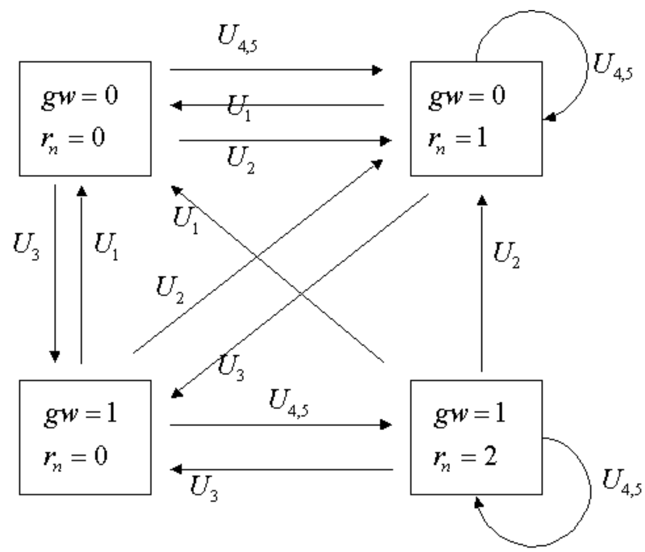

Figure 2 -- 3 node state machine to determine neighborhood best

With this enhancement, the problem of 2 clustered nodes maintaining links with a distance node is alleviated, as one of the nodes becomes a gateway and the non-gateway node disconnects from the distant node and reduces its communication range based on a revised neighborhood best $r_{i,nbest}$.

5. SIMULATION AND RESULTS

A simulation environment, using C++ was constructed to allow us to place nodes either constructively or randomly, assign initial communication ranges and execute the algorithm. Nodes are created in a 100x100 unit rectangle. The maximum communication range is set at 100 and the minimum communication range, $r_{\min}$ is set such that for the experimental choices of $c_1, c_2, \alpha, \beta, \gamma,$ *and* δ, the fitness of a node cannot improve by reducing its communication range if that reduction causes the node's 'reachability' to decrease. If the distance between a node and its nearest neighbor is less than this minimum $r_{\min}$, then the node can sacrifice 'reachability' and end up with a better fitness. To illustrate this concept, Figure 3 presents an initial and final configuration of a set of three nodes. There could be many other nodes in the environment, but these 3 are simply not connected to the rest in these 2 instances.

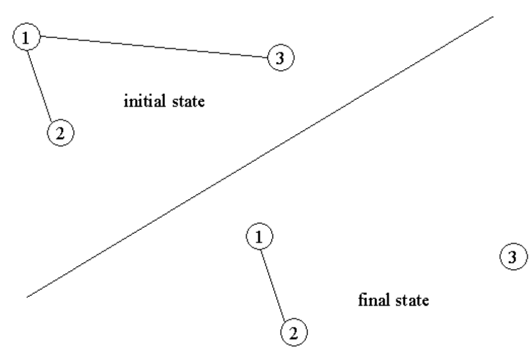

Figure 3 – Node 1 adopts a smaller communication range at the expense of connectivity.

In Figure 3, the initial state for node 1 is represented by the values,

$$\begin{aligned} S_{initial} &= 3 \\ N_{1,initial} &= 2 \\ N_{2,initial} &= 0 \;\; (by\ construction) \\ r_{initial} &= r_{13} \end{aligned}.$$

In the final configuration,

$$\begin{aligned} S_{final} &= 2 \\ N_{1,final} &= 1 \\ N_{2,final} &= 0 \;\; (by\ construction) \\ r_{final} &= r_{12} \end{aligned}.$$

To discourage nodes from making transitions like the one illustrated in Figure 3, we must force the fitness of the final state to be larger than the fitness of the initial, more highly connected, state. Therefore,

$$f_{initial} < f_{final},$$

or given that $c_1 = 0, c_2 = 1, \alpha = 1, \beta = 1, \gamma = 1$ and allowing δ as a free parameter,

$$\frac{r_{13}}{3^\delta} < \frac{r_{12}}{2^\delta}.$$

Rearranging and identifying that in the worst case, $r_{13} = r_{\max} = 100$, we find that

$$r > 100\left(\frac{2^\delta}{3^\delta}\right).$$

If $\delta = 7$, then $r_{\min} \simeq 5.85$. This bound on the minimum communication radius is an appreciable fraction of the size of the simulation environment. To allow nodes to have smaller communication radii, we either need to increase δ or decrease γ. We choose to set $\gamma = 1/2$ because the value for δ is already quite large. With these choices, the minimum communication radius is $r_{\min} \simeq .34$.

The initial communication ranges of the nodes could be set to all minimum values, all maximum values or a mixture between the two extremes. The choice of these initial values impacts the simulation results. For these investigations, we choose to initialize the communication range of the nodes to their maximum value. For the maximum communication range we have chosen, even for small numbers of nodes (3 or more), the probability that the graph is connected when all nodes transmit at their maximum range is high. [14] The PSO algorithm requires a choice for the weighting of the cognitive and social components of the particle's motion. The 'off-the-shelf' [15] PSO indicates that 2.8 and 1.3 are reasonable choices for the weighting of the cognitive and social components, respectively, but based on our experimentations, we adopt values of 1.75 and .35. Because the form of the fitness is such that larger radii are desirable for disconnected nodes while smaller radii are desirable for connected nodes, oscillations may occur. We use a random variable to weigh the node's positional motion in Eqn. 10 to quench these oscillations. We implemented a 'fitness timer' to expire periodically for each node, which effectively restarted the node's search for a best communication range. Allowing a node to forget about a previous best communication range helps a node to maintain gateway status in the event that it was previously a part of a connected network and had a smaller communication range. We executed experiments with 2, 3, 4 and 5 nodes to expose whether or not the distributed PSO algorithm offers any promise of approximating the solution to the problem stated and to assist in choosing good values for simulation parameters. The experiments used values for $c_1, c_2, \alpha, \beta, \gamma,$ *and* δ of 0,1,1,1,0.5 and 7 respectively. The base parameters of the investigated DPSO algorithm for this problem are summarized in Table 4.

Table 4 – base simulation parameters

$c_1, c_2, \alpha, \beta, \gamma,$ *and* δ	0,1,1,1,0.5 and 7 respectively
ϕ_1	distributed on [0,1.75]
ϕ_2	distributed on [0,0.35]
$r_{\min}$	.34
$r_{\max}$	100
range of x	$-50 < x < 50$
range of y	$-50 < y < 50$
initial swarm particle velocities	randomly and uniformly distributed on [-5,5]
$v_{\max}$ (heuristically constricted)	$5-4*(i/I)$, where i is current iteration and I is stopping iteration, taken to be 700
initial communication range	set to $r_{\max}$

5.1 2-Node Experiments

With the formulation of the neighborhood best from section 4.2.1, call it the 'basic' $r_{i,nbest}$ formulation, the DPSO algorithm can easily solve the problem for 2 nodes as shown in Figure 4.

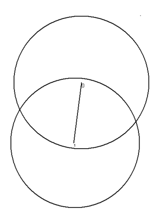

Figure 4 –2 node steady state, 'basic' $r_{i,nbest}$

5.2 3-Node Experiments

The DPSO algorithm, with the 'gateway enhanced' $r_{i,nbest}$ of section 4.2.2 is able to consistently converge to the optimal solution for 3 nodes. A typical result is displayed in Figure 5.

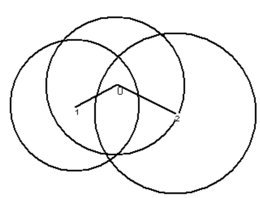

Figure 5 -- 3 node steady state, 'gateway enhanced' $r_{i,nbest}$

5.3 4 and 5 Node Experiments

When simulating with 4 nodes and more, we introduce more structure into the FSM that is used to determine the social component of the node's motion. Environmental states added to the structure include, the distance to the nearest node contributing to an asymmetric link and the distance between a node's nearest gateway and its nearest node contributing to an asymmetric link. We also expand $r'_{i,nbest}$ by one state and allow a node to set its neighborhood best communication radius equal to the distance to the nearest node contributing to an asymmetric link. The benefit derived from introducing these modifications was discovered through experimentation. The actual FSM used for 4 and 5 node simulations is rather large and is not included here. While checking 3 node simulations for optimality is trivial, doing so for 4 or more nodes requires a check by a complete algorithm that guarantees convergence to the optimal. Since 4 and 5 node configurations are still small enough to allow an exhaustive search for the optimal solution, we implemented an exhaustive search in Matlab.

Ten sets of 4 nodes and 10 sets of 5 nodes are randomly generated. For each set, the optimal communication ranges subject to the constraints are found using exhaustive search. The DPSO algorithm is executed for each set of nodes and the results tabulated. For the 10 sets of 4 nodes, we find that in 8 of the 10 sets, the DPSO algorithm finds the optimal solution. In the other 2 sets, the DPSO algorithm finds a lower cost topology at the expense of 1 asymmetric link. In the 10, 5-node sets, the DPSO algorithm finds the optimal topology in 3 out of 10, a lower cost topology at the expense of, on average, 1.5 asymmetric links, and a higher cost topology with, on average, 1.34 asymmetric links in 3 of the sets. Only one trial of the DPSO algorithm is executed for each node set and compiling more statistical data is for future work. No attempt is made to optimize the FSM developed for 4 nodes when simulating with 5. We suspect that optimization of the FSM could improve the 5 node results.

6. SUMMARY AND FUTURE WORK

The results are promising. The algorithm performs well for 2,3 and 4 node configurations. For 2 and nodes, as long as the nodes are within range, the algorithm works without fail. For 4 nodes, the algorithm converges to the optimal solution in 70% of our trials. The algorithm found a lower cost topology in the remaining 30% of trials at the expense of only a single asymmetric link. The algorithm performed well in 5 node trials even though no attempt was made to improve the FSM used to determine the neighborhood best. Gains could be made in improving the results for 5 and possibly more nodes by expanding the state space of the FSM and optimizing the transition function off-line to a global fitness criterion. The DPSO parameters ϕ_1, ϕ_2 could be included in the off-line optimization. It may be possible to evolve these parameters on-line and hence make an adaptive algorithm that could be apart of a general-purpose and reusable architecture for networking wireless ad-hoc nodes.

One clear limitation of the algorithm is that it does not explicitly search for a global optimum. As mentioned in the previous paragraph, a global tendency could be engineered through the choice of the FSM and DPSO parameters, yet the algorithm still proceeds with nodes operating in a selfish mode otherwise. This selfish behavior could be softened through incorporating another term in to the node's fitness function, which includes a neighborhood component. With such a fitness representation, nodes may choose a poorer value for the selfish component of their fitness in order to benefit the group, or their local neighborhood.

The true test of a topology control algorithm lies in simulation. The application ultimately determines whether or not a given configuration is optimal.[1] The topology generated by the algorithm should be simulated in the Network Simulator[16] and compared to results of other topology generation schemes. [12, 17, 18] Of particular interest is determining the topological characteristics that make a given topology a good topology. While most existing approaches are typically inflexible, the algorithm presented here offers the possibility of adapting and conforming to operational parameters that do indeed optimize the performance of the wireless network.

REFERENCES

[1] S.-J. Park and R. Sivakumar, "Adaptive Topology Control for Wireless Ad-hoc Networks," presented at MobiHoc, Annapolis MD, 2003.

[2] V. Bharghavan, A. Demers, S. Shneker, and L. Zhang, "Macaw: A media access protocol for wireless lans," presented at Proc. ACM SIGCOMM '94, 1994.

[3] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "Analysis of a Cone-Based Distributed Topology Control Algorithm for Wireless Multi-hop Networks," presented at ACM Principles of Distributed Computing, 2001.

[4] N. A. Lynch, "Distributed Algorithms," Morgan Kaufmann Publishers, 1996, pp. 51-80.

[5] S. M. Merritt, "A top down unification of minimum cost spanning tree algorithms," presented at Symposium proceedings on Communications architectures & protocols, Austin, Texas, United States, 1989.

[6] C. E. Perkins, E. M. Royer, and S. Das, "Ad hoc on demand distance vector routing," presented at 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999.

[7] C. E. Perkins and P. R. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers," presented at ACM SIGCOMM, 1994.
[8] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Standard 802.11," IEEE 802 LAN/MAN Standards Committee 1999.
[9] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. IT-46, pp. 388-404, 2000.
[10] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol," presented at European Wireless Conference, 2002.
[11] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*: Morgan Kaufmann Publishers, 2001.
[12] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed Topology Control for Ad-hoc Networks," presented at IEEE INFOCOM, 2001.
[13] S. Agarwal, S. Krishnamurthy, R. H. Katz, and S. K. Dao, "Distributed power control in ad hoc wireless networks," presented at PIMRC01 -- IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2001.
[14] B. Krishnamachari, S. Wicker, R. Bejar, and M. Pearlman, "Critical Density Thresholds in Distributed Wireless Networks," in *Communications, Information and Network Security*: Kluwer Publishers, 2002.
[15] A. Carlisle and G. Dozier, "An off-the-shelf PSO," presented at Workshop on Particle Swarm Optimization, Indianapolis, IN: Purdue School of Engineering and Technology, 2001.
[16] UCB/LBNL/VINT, "The Network Simulator (ns-2)," 2000.
[17] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," presented at INFOCOM 2000, 2000.
[18] V. Rodoplu and T. H. Meng, "Minimum Energy Mobile Wireless Networks," presented at IEEE International Conference on Communications, ICC '98, 1998.